Automated Discovery of Concurrent Models of Decision-Making Systems from Data

Zbigniew Suraj* ORCID: 0000-0001-9544-9561, Piotr Grochowalski ORCID: 0000-0002-4124-412X, Paweł Drygaś ORCID: 0000-0001-6954-5971

Institute of Computer Science, University of Rzeszów
35-310 Rzeszów, 1 Prof. S. Pigonia Str., Poland
e-mail: {zsuraj,pgrochowalski,padrygas}@ur.edu.pl

Abstract. The paper presents a methodology for building concurrent models of decision-making systems based on knowledge extracted from empirical data. We assume that the data is represented by a decision table, while the decision-making system is represented by a Petri net. Decision tables contain conditional attribute values obtained from measurements or other sources. A Petri net is constructed using all true and acceptable rules generated from a given decision table. Rule factors and other parameters needed to build the net model are also computed from the data table. Three operators In, Trs and Out interpreted as uninorms are used to describe the dynamics of the net model. The expected behavior of the model is achieved by proper organization of its work. The theoretical basis of the methodology is the concepts, methods and algorithms derived from the theory of rough sets, fuzzy sets and Petri nets.

Key words: Rough set; Fuzzy set; Petri net; Decision rule; Uninorm

1 Introduction

One of the challenges of modern artificial intelligence [1] and IT is building intelligent systems that can function reliably and efficiently, also in uncertain conditions. It seems that in order to satisfactorily meet this task, it is no longer enough to use the achievements of science in one field. In this paper, we try to solve the problem posed in it, using an approach based on three widely recognized theories: rough sets, fuzzy sets and Petri nets. Rough set theory is an important mathematical and artificial intelligence technique proposed by Pawlak in 1982 [2]. The main feature of this theory, from the point of view of practical applications, is the classification of empirical data, and thus decision making. Rough sets and fuzzy sets are quite closely related. The latter were introduced in 1965 by Zadeh [3]. Both of these concepts can be regarded as generalizations of a set in the classical sense. In turn, Petri nets were introduced by Petri in 1962 [4] as one of the formalisms used to model and analyze the behavior of systems of concurrent processes.

* Corresponding author

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_15

In the paper, we assume that empirical knowledge about the modeled system is presented in the form of a given decision table in the sense of Pawlak [5]. The decision table DT consists of a series of rows labeled by the names of elements from the set of objects U, represented by a vector of conditional attribute values from the set A along with the corresponding decision d. The input data for the construction of a decision-making system model are rules generated from a given decision table. The resulting system model is represented by a weighted priority uninorm Petri net (WPUP-net), which enables decision making as soon as there are enough values of conditional attributes represented by the so-called starting places in the net model, based on the knowledge encoded in DT (cf. [6]). We consider two types of rules generated from DT: true and acceptable. A rule is true in DT if and only if any object u in DT that matches the left-hand side of the rule also matches its right-hand side, and there is an object u that matches its left-hand side. A rule is acceptable in DT if the match of any object to the rule is not exact, but only to some non-zero degree (cf. [7]). A rule is active when values are specified for all attributes on its left-hand side. We assume that the net model proposed in this paper will transfer information from one attribute to another as quickly as possible. Therefore, there is a need to generate both all true and acceptable rules from DT [6]. Finally, our net model is an implementation of a set of generated rules and their parameters using WPUP-nets. The proposed net model works as follows. There are two phases in every net computation. In the first phase it is checked whether the values of any conditions are known, if so, the net tries to run decision rules, if possible, if not, then in the second phase the net tries to generate new information about the condition values and send them across the net. The entire computational process is carried out through the appropriate organization of the net operation based on the prioritization of transitions. Transitions representing conditional rules have a lower priority than transitions representing decision rules. This gives the desired effect.

It is worth emphasizing that in our approach, in addition to calculating rules from a given decision table, all rule coefficients and other parameters needed to build a Petri net model are also calculated from it, and they do not come from an expert in a given field of application. This aspect clearly distinguishes our approach from other approaches commonly available in the literature. This also means that the process of modeling decision-making systems using our methodology can be largely automated, not only modeling, but also analyzing and verifying the correctness of its operation using specialized software such as PNeS (Petri Net System), which is designed to assist users in modeling and analyzing systems of concurrent processes with different types of Petri nets [8]. In 1984, Lipp [9] published the first paper on fuzzy Petri nets, which work well in modeling systems operating in imperfect information environments. Since their introduction, they have enjoyed unflagging interest among people dealing with artificial intelligence and computer science [10]. As indicated in the review literature [11],[12], these nets have some disadvantages and therefore are not fully suitable for modeling complex decision systems. For this reason, there are many alternative models in the literature that increase both the ability to represent

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_15

knowledge more accurately and the ability to reason more intelligently. To the best of our knowledge, we are not aware of studies using the Petri nets based on uninorms that fit into the above-mentioned research trend. Uniforms were introduced by Yager and Rybalov in 1996 [13] as a generalization of t-norms and t-conorms, almost universally used to describe the dynamics of fuzzy Petri nets. With uninorms, logical connectives AND and OR can be modeled more adequately than using triangular norms. By manipulating the values of the neutral element in uninorms, such an effect is obtained. Since their introduction, uninorms have been used in expert systems [14],[15], decision systems [16],[17], and others. In this paper, uninorms are used both to represent knowledge and to describe the decision-making process implemented in WPUP-net models.

The work is theoretical in nature and contains potential applications in modeling decision-making systems operating in an uncertain environment. We believe this is an important extension of the research described in [18]. This extension applies, among others: (1) adding information about uninorms; (2) development of a new Petri net model based on uninorms, not triangular norms; (3) modified description of both the structure and behavior of the new net model; (4) a modified net representation of rule knowledge, in which the In, Trs and Out operators are now uninorms, thanks to which the logical operators And and Or appearing in the rules can be better modeled in the environment of uncertain information; (5) modification of the operating algorithm of the developed net model; (6) building and analyzing the operation of a net model of a decision-making system described by a decision table based on a new type of Petri net, from which decision and conditional rules were extracted along with the necessary parameter values needed to automatically build such a model. We consciously use similar examples in the introductory part of this work, as well as in the main example illustrating our approach to help the potential reader see the similarities and differences between the current methodology and that described in [18].

The rest of this paper is structured as follows. Sect. 2 recalls the basic concepts and notations for rough sets and uninorms, and illustrates them with examples. In Sect. 3, a new model of Petri nets based on uninorms is presented. Sect. 4 contains net representations of rules. In Sect. 5, an algorithm that describes how the WPUP-net should work is introduced. Sect. 6 gives an example to illustrate our methodology. In Sect. 7, conclusions and directions for further work are presented.

2 Backgrounds and Examples

2.1. Rough sets

A pair $S = (U, A)$ is called an *information system* if U is a nonempty finite set of objects, called the *universe*, A is a nonempty finite set of *attributes* and $a : U \to V_a$ for every $a \in A$. The set V_a is called the *value set* of a, and $V = \bigcup_{a \in A} V_a$ is called the *domain* of A.

A *decision table* is a pair $DT = (U, A \cup \{d\})$, where A is a nonempty set of *conditional attributes*, $d \notin A$ is a *decision attribute* (*decision*). Any decision

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_15

table $DT = (U, A \cup \{d\})$ can be represented by a table with a number of rows equal to the size of the universe U and a number of columns equal to the size of the set $A \cup \{d\}$. The value $a(u)$ appears at the position corresponding to row u and column a.

Let $S = (U, A)$ be an information system, $B, C \subseteq A$. A set C is dependent to *degree* k on B in S if $k = \gamma(B, C) = \frac{|POS_B(C)|}{|U|}$, where the set $POS_B(C)$ is the positive region of the partition U/C w.r.t. B, i.e., it is the set of all elements of U that can be uniquely classified to blocks of the partition U/C by means of B [22]. If $k = 1$ a set C is *totally* dependent on B, if $k = 0$ a set C is *totally* independent on B and otherwise C is *roughly* dependent on B.

Let $B, C \subseteq A$, and $B' \subseteq B$. We say that a set B' *is a relative reduct of* B *w.r.t.* C, if B' is a minimal subset of B and $\gamma(B, C) = \gamma(B', C)$.

Let $DT = (U, A \cup \{d\})$ be a decision table, $B \subseteq A$. We consider two natural coefficients of the significance based on the degree of dependency between the attribute sets B and $\{d\}$:

$$\sigma_1(B, d, a) = \gamma(B, \{d\}) - \gamma(B - \{a\}, \{d\}) = \frac{|POS_B(\{d\})| - |POS_{B-\{a\}}(\{d\})|}{|U|},$$

$$\sigma_2(B, d, a) = \frac{\gamma(B,\{d\}) - \gamma(B-\{a\},\{d\})}{\gamma(B,\{d\})} = \frac{|POS_B(\{d\})| - |POS_{B-\{a\}}(\{d\})|}{|POS_B(\{d\})|},$$

and denoted by $\sigma_1(a)$ ($\sigma_2(a)$), when B and $\{d\}$ are understood. σ_1 measures the difference between $\gamma(B, \{d\})$ and $\gamma(B - \{a\}, \{d\})$, i.e., it determines how the value of $\gamma(B, \{d\})$ changes after removing the attribute a, whereas σ_2 normalizes this difference. In practice, the more important the attribute a, the greater the value of both coefficients. It is true that: $0 \leq \sigma_1(a) \leq \sigma_2(a) \leq 1$.

Let $DT = (U, A \cup \{d\})$ be a decision table, $B \subseteq A \cup \{d\}$, and $V = \bigcup_{a \in A} V_a \cup V_d$. Expressions of the form $a = v$, where $a \in B$ and $v \in V_a$ are called *descriptors* over B and V. By $\mathrm{DESC}(B, V)$, we denote the set of all descriptors over B and V which is the smallest such set and closed w.r.t. classical connectives: OR (disjunction), AND (conjunction) and NOT (negation).

Let $\tau \in \mathrm{DESC}(B, V)$. The set of all objects in U with property τ is called the *meaning* of τ in the decision table DT and denoted by $\parallel \tau \parallel$.

Let DT be a decision table and $\mathrm{DESC}(A, V_a)$, $a \in A$ be the set of *conditional formulas of* DT. Any expression of the form $\tau \to d = v$, where $\tau \in \mathrm{DESC}(A, V_a)$, $v \in V_d$ and $\parallel \tau \parallel \neq \emptyset$ is called a *decision rule* r in DT. The formula τ is called the *predecessor* and the formula $d = v$ *successor* of the decision rule r. A non-empty set of objects $\parallel \tau \parallel$ is *matching* the decision rule, and the set of objects $\parallel \tau \parallel \cap \parallel (d = v) \parallel$ is *supporting* the rule. By *accuracy factor* of the decision rule r we mean the number $acc(r) = \frac{|\parallel \tau \parallel \cap \parallel (d=v) \parallel|}{|\parallel \tau \parallel|}$, whereas by *coverage factor* of r the number $cov(r) = \frac{|\parallel \tau \parallel \cap \parallel (d=v) \parallel|}{|\parallel (d=v) \parallel|}$. The *strength factor* of the decision rule r is the number $str(r) = \frac{|\parallel \tau \parallel \cap \parallel (d=v) \parallel|}{|U|}$. We say that a decision rule r is *true* in DT, if $acc(r) = 1$, and it is *acceptable* in DT, if $0 < acc(r) < 1$. A decision rule r is called *minimal* if it has the minimum number of descriptors on the left-hand side. It is obvious that: $0 \leq str(r) \leq acc(r) \leq 1$ and $0 \leq str(r) \leq cov(r) \leq 1$ for every decision rule r in DT.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_15

Remark. (1) All the terms defined above for decision rules also apply to conditional rules of the form: $\tau \to a = v$, where $\tau, a = v \in \mathrm{DESC}(A, V_a)$. (2) In this paper, $\sigma_2(a)$, $acc(r)$ and $cov(r)$ are the parameters used to characterize the rules (see Section 4).

Table 1. A decision table

U / $A \cup \{d\}$	H	M	T	F
u_1	no	yes	high	yes
u_2	yes	no	high	yes
u_3	yes	yes	very high	yes
u_4	no	yes	normal	no
u_5	yes	no	high	no
u_6	no	yes	very high	yes

Example 1. Consider the decision table DT from table 1. We have: $U = \{u_1, u_2, u_3, u_4, u_5, u_6\}$; $A = \{\mathsf{H}, \mathsf{M}, \mathsf{T}\}$, where H is headache, M - muscle pain, T - temperature; $d = \mathsf{F}$ and means flu. The attribute d represents the expert's decision to diagnose flu based on the patient's interview. Attribute values from $A \cup \{d\}$ are presented inside the table. We can calculate: $\gamma(\{\mathsf{H}, \mathsf{M}, \mathsf{T}\}, \{\mathsf{F}\}) = \frac{2}{3}$, $\gamma(\{\mathsf{T}\}, \{\mathsf{F}\}) = \frac{1}{2}$, $\gamma(\{\mathsf{H}\}, \{\mathsf{F}\}) = \gamma(\{\mathsf{M}\}, \{\mathsf{F}\}) = 0$, and two relative reducts w.r.t. $\{\mathsf{F}\}$, R_1=$\{\mathsf{H},\mathsf{T}\}$ and R_2=$\{\mathsf{M},\mathsf{T}\}$ of the set of conditions $\{\mathsf{H},\mathsf{M},\mathsf{T}\}$. Using the formulas for σ_1 and σ_2 for Table 1, we obtain the following measures of the significance of some attributes from the set A w.r.t. the classification generated by: (1) the conditional attributes A: $\sigma_1(\mathsf{H}) = 0$, $\sigma_2(\mathsf{H}) = 0$, $\sigma_1(\mathsf{M}) = 0$, $\sigma_2(\mathsf{M}) = 0$, $\sigma_1(\mathsf{T}) = \frac{1}{2}$, $\sigma_2(\mathsf{T}) = \frac{3}{4}$; (2) the relative reduct R_1: $\sigma_1(\mathsf{H}) = \frac{1}{6}$, $\sigma_2(\mathsf{H}) = \frac{1}{4}$, $\sigma_1(\mathsf{T}) = \frac{2}{3}$, $\sigma_2(\mathsf{T}) = 1$; (3) the relative reduct R_2: $\sigma_1(\mathsf{M}) = 0$, $\sigma_2(\mathsf{M}) = \frac{1}{4}$, $\sigma_1(\mathsf{T}) = \frac{3}{4}$, $\sigma_2(\mathsf{T}) = 1$. Furthermore, using the method of generating the minimal rules in DT [6], we obtain the following rules along with a list of numerical factors corresponding to:

1. Nontrivial functional dependencies between the values of conditions T and H in the reduct R_1: $r_1 :=$ (T=very high) $\to$ (H=no); $[\sigma_2(\mathsf{T}) = 1$, $cov(r_1) = \frac{1}{3}$ / $str(r_1) = \frac{1}{6}$; $acc(r_1) = \frac{1}{2}]$; $r_2 :=$ (T=very high) $\to$ (H=yes); $[\sigma_2(\mathsf{T}) = 1$, $cov(r_2) = \frac{1}{3}$ / $str(r_2) = \frac{1}{6}$; $acc(r_2) = \frac{1}{2}]$; $r_3 :=$ (T=high) $\to$ (H=no); $[\sigma_2(\mathsf{T}) = 1$, $cov(r_3) = \frac{1}{3}$ / $str(r_3) = \frac{1}{6}$; $acc(r_3) = \frac{1}{3}]$; $r_4 :=$ (T=high) $\to$ (H=yes); $[\sigma_2(\mathsf{T}) = 1$, $cov(r_4) = \frac{2}{3}$ / $str(r_4) = \frac{1}{3}$; $acc(r_4) = \frac{2}{3}]$; $r_5 :=$ (T=normal) $\to$ (H=no); $[\sigma_2(\mathsf{T}) = 1$, $cov(r_5) = \frac{1}{3}$ / $str(r_5) = \frac{1}{6}$; $acc(r_5) = 1]$.

2. Nontrivial functional dependencies between the values of conditions from R_1=$\{\mathsf{H},\mathsf{T}\}$ and the decision F: $r_6 :=$ (T=very high) $\to$ (F=yes); $[\sigma_2(\mathsf{T}) = 1$, $cov(r_6) = \frac{1}{4}$ / $str(r_6) = \frac{1}{6}$; $acc(r_6) = 1]$; $r_7 :=$ (H=no) AND (T=high) $\to$ (F=yes); $[\sigma_2(\mathsf{H}) = \frac{1}{4}$, $\sigma_2(\mathsf{T}) = 1$, $cov(r_7) = \frac{1}{4}$ / $str(r_7) = \frac{1}{6}$; $acc(r_7) = 1]$; r_8 := (H=yes) AND (T=high) $\to$ (F=yes); $[\sigma_2(\mathsf{H}) = \frac{1}{4}$, $\sigma_2(\mathsf{T}) = 1$, $cov(r_8) = \frac{1}{4}$ / $str(r_8) = \frac{1}{6}$; $acc(r_8) = \frac{1}{2}]$; $r_9 :=$ (H=yes) AND (T=high) $\to$ (F=no); $[\sigma_2(\mathsf{H}) =$

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_15

$\frac{1}{4}$, $\sigma_2(\mathsf{T}) = 1$, $cov(r_9) = \frac{1}{2}$ / $str(r_9) = \frac{1}{6}$; $acc(r_9) = \frac{1}{2}$]; $r_{10} :=$ (T=normal) $\rightarrow$ (F=no); [$\sigma_2(\mathsf{T}) = 1$, $cov(r_{10}) = \frac{1}{2}$ / $str(r_{10}) = \frac{1}{6}$, $acc(r_{10}) = 1$]. Note that r_5, r_6, r_7 and r_{10} are true in Table 1, while the rest are only acceptable.
Remark. These rules can also be generated from Table 1 using e.g. PNeS [8].

2.2. Uninorms

A mapping $U: [0,1]^2 \rightarrow [0,1]$ is called a *uninorm* if it is commutative, associative, nondecreasing, and there exists $e \in [0,1]$ (called neutral element) such that $U(e,x) = x$ for all $x \in [0,1]$.

The function U becomes a t-norm when $e = 1$ and an s-norm (t-conorm) when $e = 0$. Both classes of triangular norms are commonly used in fuzzy logic [19]. It is true that for every $(x,y) \in [0,e) \times (e,1] \cup (e,1] \times [0,e)$, $min(x,y) \leq U(x,y) \leq max(x,y)$. Moreover, $U(0,1) \in \{0,1\}$ for all uninorms U [20]. If $U(0,1) = 0$, then the uninorm U is called *andlike* (or *conjunctive*), and if $U(0,1) = 1$, then U is called *orlike* (or *disjunctive*).

Fact. If U is a uninorm with $e \in (0,1)$ and the functions $x \rightarrow U(x,1)$ and $x \rightarrow U(x,0)$ ($x \in [0,1]$) are continuous, except perhaps $x = e$. Then U can be determined using one of the formulas below.

(a) If $U(0,1) = 0$ then

$$U(x,y) = \begin{cases} eT_U(\frac{x}{e}, \frac{y}{e}) \text{ if } (x,y) \in [0,e]^2 \\ e + (1-e)S_U(\frac{x-e}{1-e}, \frac{y-e}{1-e}) \text{ if } (x,y) \in [e,1]^2 \\ min(x,y) \text{ in other cases} \end{cases} \quad (1)$$

(b) If $U(0,1) = 1$ then

$$U(x,y) = \begin{cases} eT_U(\frac{x}{e}, \frac{y}{e}) \text{ if } (x,y) \in [0,e]^2 \\ e + (1-e)S_U(\frac{x-e}{1-e}, \frac{y-e}{1-e}) \text{ if } (x,y) \in [e,1]^2 \\ max(x,y) \text{ in other cases} \end{cases} \quad (2)$$

In both formulas given above, T_U is a t-norm and S_U is an s-norm. We denote the class of uninorms of the form (1) by U_{min}, and the class of uninorms of the form (2) by U_{max}. The above relationships allow us to determine general formulas for uninorms with a neutral element e. In this paper, we limit our considerations to six uninorms, simple in the mathematical notation, which general formulas along with names borrowed from the names of the appropriate triangular norms used to determine them are given in Tables 2 and 3.

Table 2. Uninorms with $e \in (0,1)$ corresponding to three basic t-norms [21]

Name	*Formula for* $x, y \in [0,e]$
Zadeh t-uninorm	$U^e_{Z_t}(x,y) = min(x,y)$
Goguen t-uninorm	$U^e_{G_t}(x,y) = \frac{xy}{e}$
Lukasiewicz t-uninorm	$U^e_{L_t}(x,y) = max(0, x+y-e)$

Example 2. Let $U \in U_{min}$ and $U' \in U_{max}$ be uninorms defined as follows:

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_15

Table 3. Uninorms with $e \in (0, 1)$ corresponding to three basic s-norms [21]

$Name$	$Formula\ for\ x, y \in [e, 1]$
Zadeh s-uninorm	$U^e_{Z_s}(x, y) = max(x, y)$
Goguen s-uninorm	$U^e_{G_s}(x, y) = \frac{x+y-xy-e}{1-e}$
Lukasiewicz s-uninorm	$U^e_{L_s}(x, y) = min(1, x + y - e)$

$$U(x, y) = \begin{cases} max(0, x + y - 1/2) \text{ if } (x, y) \in [0, 1/2]^2 \\ min(1, x + y - 1/2) \text{ if } (x, y) \in [1/2, 1]^2 \\ min(x, y) \text{ in other cases} \end{cases}$$

$$U'(x, y) = \begin{cases} max(0, x + y - 1/2) \text{ if } (x, y) \in [0, 1/2]^2 \\ min(1, x + y - 1/2) \text{ if } (x, y) \in [1/2, 1]^2 \\ max(x, y) \text{ in other cases} \end{cases}$$

Note that U and U' are andlike and orlike uninorms, respectively. Moreover, both formulas can be obtained from Lukasiewicz's t-norm and s-norm shown in Tables 2 and 3, respectively, with the neutral element $e = 1/2$. For more information on rough sets and uninorms, see [22],[23].

3 Weighted Priority Uninorm Petri Nets

Let U_{min} and U_{max} denote the classes of uninorms of the form (1) and (2), respectively, defined in Section 2, with the neutral element $e \in (0, 1)$.

A tuple $N_U = (P, T, I, O, Pr, M_0, S, \alpha, \beta, \gamma, Op, \delta)$ is called a *weighted priority uninorm Petri net* (WPUP-net for short) if: (1) $P = \{p_1, p_2, ..., p_n\}$ is a set of places, $T = \{t_1, t_2, ..., t_m\}$ is a set of transitions; (2) $I: P \times T \to [0, 1]$ is the input function, $O: T \times P \to [0, 1]$ is the output function; (3) $Pr: T \to \mathbb{N}$ is the priority function ($\mathbb{N}$ is the set of natural numbers), $M_0: P \to [0, 1]$ is the initial marking; (4) $S = \{s_1, s_2, \ldots, s_n\}$ is a set of statements (P, T, S are pairwise disjoint), $\alpha: P \to S$ is the statement binding function; (5) $\beta: T \to [0, 1]$ is the truth degree function, $\gamma: T \to [0, 1]$ is the threshold function; (6) $Op = U_{min} \cup U_{max}$ is the set of operators, $\delta: T \to Op \times Op \times Op$ is the operator binding function.

Let N_U be a WPUP. For $t \in T$: ${}^\bullet t = \{p : I(p, t) > 0\}$ is a set of input places of t, and $t^\bullet = \{p' : O(t, p') > 0\}$ is a set of all output places of t.

By tokens we mean the values of the function M_0. We assume that the set O_p contains uninorms. If the arc (p, t) connects p and t, then $I(p, t) > 0$, otherwise 0. The value $I(p, t)$ is called the input weight t and is denoted iw. Similarly, if the arc (t, p) connects t and p, then $O(t, p) > 0$, otherwise 0. The value $O(t, p)$ is called the output weight t and is denoted ow. Places are graphically represented by circles and transitions by rectangles. If the weight of the directed arc is 1, then 1 is not shown in the net graph, if the weight of the directed arc is 0, this arc is skipped in the net graph. In other cases, the weight is placed next to the arc in

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_15

question. Priorities are placed next to transitions. We only consider two priority values: 0 and 1. If the priority is 0, we do not show it in the net graph. Transitions representing decision rules have a priority of 1, all others have a priority of 0. Each place contains one token. The token is placed inside the place. If a token has a value of 0 in a given place, then that place is empty. Each statement is associated with only one place. The statement is placed next to the place. The values of $\beta(t)$ and $\gamma(t)$ are shown in the net graph under the transition t. The first value is interpreted as the accuracy factor of the rule represented by t. The second value affects the activation of the transition. If the value of the transition firing condition is not less than the threshold value, the transition can be fired. The value of the function δ is the triple of operators (In, Trs, Out), this triple is placed under the transition. In is the input operator, Trs is the transmission operator, and Out is the output operator. Operator In aggregates tokens placed in the input places of the transition to which it is assigned. The role of the Trs and Out operators is to aggregate the value received from In with the values of the remaining net parameters and send the generated value to all transition output places to which these three operators are assigned. We assume that the input operators belong to U_{min} or U_{max}, while the other two belong to U_{min} and U_{max}, respectively. These operators are used to describe the dynamics of the WPUP net. Arc weight values and β function values are calculated from the data table and interpreted using rough set theory concepts (see Section 4). However, values of the threshold function γ are set by the domain expert.

WPUP-net dynamics are defined by a *firing rule*, and the net evolution is represented by a *sequence of fired transitions*.

Let N_U be a WPUP-net. A marking of N_U is a function $M \colon P \to [0, 1]$.

Firing rule. Let $N_U = (P, T, S, I, O, \alpha, \beta, \gamma, Op, \delta, M_0)$ denotes a WPUP-net, M be a marking of N_U, $t \in T$, ${}^\bullet t = \{p_{i1}, p_{i2}, \ldots, p_{ik}\}$ be a set of input places for t, $\beta(t) \in (0, 1]$, and $\delta(t) = (In, Trs, Out)$. A transition t is *enabled* (or *ready to fire*) for marking M if for all $p \in {}^\bullet t$: $In((iw_{i1} \cdot M(p_{i1}), iw_{i2} \cdot M(p_{i2})), ..., iw_{ik} \cdot M(p_{ik})) \geq \gamma(t) > 0$, where iw_{ij} is an input weight of an arc (p_{ij}, t) and $M(p_{ij})$ is a marking of a place p_{ij} for $j = 1, 2, ..., k$.

If ${}^\bullet t$ consists of only one place p, we assume that a transition t is enabled for M if the following condition is true: $iw \cdot M(p) \geq \gamma(t) > 0$, where iw is an input weight of arc (p, t). A net transition can be fired when it is enabled.

According to the definition of WPUP-net, transitions can be assigned priorities, which means that if two or more transitions are enabled simultaneously in a given marking, the transitions with the highest priority will be activated first [24].

Formula of the next marking. Firing an enabled transition t by a marking M results in a new marking M' defined by

$$M'(p) = \begin{cases} Out(ow \cdot Trs(In(iw_{i1} \cdot M(p_{i1}), iw_{i2} \cdot M(p_{i2}), ..., iw_{ik} \cdot M(p_{ik})), \beta(t)), \\ M(p)) \text{ if } p \in t^\bullet \\ M(p) \text{ otherwise} \end{cases}$$

where ow is an output weight of arc (t, p).

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_15

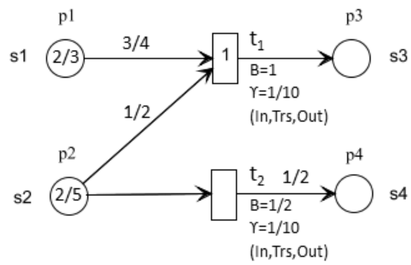

Fig. 1. A WPUP-net with the initial marking $M_0 = (2/3, 2/5, 0, 0)$.

Example 3. For the WPUP-net of Fig. 1, we have: $P = \{p_1, p_2, p_3, p_4\}$; $T = \{t_1, t_2\}$; $I(p_1, t_1) = iw_1 = 3/4$, $I(p_2, t_1) = iw_2 = 1/2$, $I(p_3, t_1) = iw_3 = 0$, $I(p_4, t_1) = iw_4 = 0$, $I(p_1, t_2) = iw_5 = 0$, $I(p_2, t_2) = iw_6 = 1$, $I(p_3, t_2) = iw_7 = 0$, $I(p_4, t_2) = iw_8 = 0$; $O(t_1, p_1) = ow_1 = 0$, $O(t_1, p_2) = ow_2 = 0$, $O(t_1, p_3) = ow_3 = 1$, $O(t_1, p_4) = ow_4 = 0$, $O(t_2, p_1) = ow_5 = 0$, $O(t_2, p_2) = ow_6 = 0$, $O(t_2, p_3) = ow_7 = 0$, $O(t_2, p_4) = ow_8 = 1/2$; $Pr(t_1) = 1$, $Pr(t_2) = 0$; $M_0 = (2/3, 2/5, 0, 0)$; $S = \{s_1, s_2, s_3, s_4\}$; $\alpha(p_1) = s_1$, $\alpha(p_2) = s_2$, $\alpha(p_3) = s_3$, $\alpha(p_4) = s_4$; $\beta(t_1) = 1$, $\beta(t_2) = 1/2$; $\gamma(t_1) = \gamma(t_2) = 1/10$; $O_p = \{In, Trs, Out\}$, where In and Trs are interpreted as uninorm U, and Out as uninorm U' from Example 2; $\delta(t_1) = \delta(t_2) = (In, Trs, Out)$. Notice that t_1 and t_2 are ready to fire by the initial marking M_0. This is because: $In(iw_1 \cdot M(p_1), iw_2 \cdot M(p_2)) = In(3/4 \cdot 2/3, 1/2 \cdot 2/5) = In(1/2, 1/5) = max(0, 1/2 + 1/5 - 1/2) = 1/5 \geq \gamma(t_1) = 1/10$ and $iw_6 \cdot M(p_2) = 1 \cdot 2/5 \geq \gamma(t_2) = 1/10$. Only t_1 will be fired because it has priority higher than priority t_2. Firing transition t_1 with the initial marking M_0 leaves this marking unchanged. This is due to the fact that: $Trs(In(iw_1 \cdot M(p_1), iw_2 \cdot M(p_2)), \beta(t_1)) = Trs(1/5, 1) = min(1/5, 1) = 1/5$ and $Out(ow_3 \cdot Trs(1/5, 1), M_0(p_3)) = Out(1 \cdot 1/5, 0) = Out(1/5, 0) = max(0, 1/5 + 0 - 1/2) = max(0, -3/10) = 0$.

Remark. Here and in the rest of the paper, instead of $\beta(t) = b, \gamma(t) = c$, where t is a transition and b, c values from the unit interval [0,1], we will use the abbreviations $\beta = b, \gamma = c$.

4 WPUP-net Representation of Rules

This section describes the three types of rules, including a list of parameters that characterize them (cf. [11],[25]).

Let $DT = (U, A \cup \{d\})$ denote a given decision table, and $\text{DESC}(A, V_a)$ be the set of its conditional formulas.

Type 1. (A simple rule.) r_1: $(a = v) \rightarrow (d = v')$ $[b;\ \sigma(a), cov(r_1);\ acc(r_1)]$ with descriptors $a = v$ and $d = v'$ such that $a = v \in \text{DESC}(A, V_a)$ and $v' \in V_d$, a truth degree value b of $a = v$, significance $\sigma(a)$ of attribute a given by the formula for $\sigma_2(a)$, a coverage factor $cov(r_1)$ and an accuracy factor $acc(r_1)$ of

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_15

rule r_1. The method for calculating these parameters can be found in Subsection 2.1.

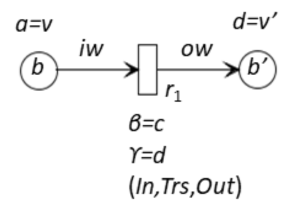

Fig. 2. A WPUP-net model of the type 1 rule after firing r_1.

The WPUP-net structure of rule r_1 is shown in Fig. 2, where $iw = \sigma(a)$ is an input weight of r_1, and $ow = cov(r_1)$ is an output weight of r_1. If iw is greater than ow, the first link is stronger than the second. However, value $c = \beta(r_1)$ is interpreted as $acc(r_1)$. As before, the higher the value of β, the more robust the rule. Value $d = \gamma(r_1)$ represents a threshold that requires degree of truth value $a = v$ to be not less than d. In, Trs, Out are operators assigned to r_1, which are the appropriate uninorms described in the WPUP-net definition (Section 3). According to Fig. 2 the value of a token at an output place of r_1 is calculated as $b' = ow \cdot Trs(b \cdot iw, c)$ if $b \cdot iw \geq d$.

If a rule's antecedent or successor contains an input operator In, it is called a *compound rule*.

Type 2. (A compound rule in the antecedent.) $r_2 : (a_1 = v_1)\ In\ (a_2 = v_2) \cdots In\ (a_k = v_k) \to (d = v')\ [b_1, b_2, \ldots, b_k;\ \sigma^1(a), \sigma^2(a), \ldots, \sigma^k(a), cov(r_2);\ acc(r_2)]$ with descriptors $a_1 = v_1$, $a_2 = v_2$, $\ldots$, $a_k = v_k$, $d = v'$ and truth degree values b_1, b_2, $\ldots$, b_k, b', respectively. A token value b' at an output place of r_2 is calculated as follows (Fig. 3): $b' = Trs(In(b_1 \cdot iw_1, b_2 \cdot iw_2, \ldots, b_k \cdot iw_k), c)) \cdot ow)$ if $In(b_1 \cdot iw_1, b_2 \cdot iw_2, \ldots, b_k \cdot iw_k) \geq d$. Note that in this case, the operator In can be interpreted as either andlike uninorm or orlike one.

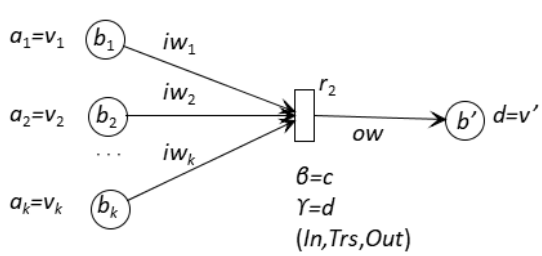

Fig. 3. A WPUP-net model of the type 2 rule after firing r_2.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_15

Type 3. (A compound rule in the successor.) r_3: $(a = v) \rightarrow (d = v_1)$ In $(d = v_2)$ $\cdots$ In $(d = v_n)$ $[b;\ \sigma^1(a), \sigma^2(a), \ldots, \sigma^n(a), cov^1(r_3), cov^2(r_3), \ldots, cov^n(r_3);\ acc^1(r_3), acc^2(r_3), \ldots, acc^n(r_3)]$ with descriptors $a = v$, $d = v_1$, $d = v_2$, $\ldots$, $d = v_n$, and a truth degree value b of $a = v$. A token value b'_j for each output place of r_3 is calculated in the following way (Fig. 4): $b'_j = ow_j \cdot Trs(b \cdot iw, c_j)$ if $b \cdot iw \geq d_j$, $j = 1, \ldots, n$.

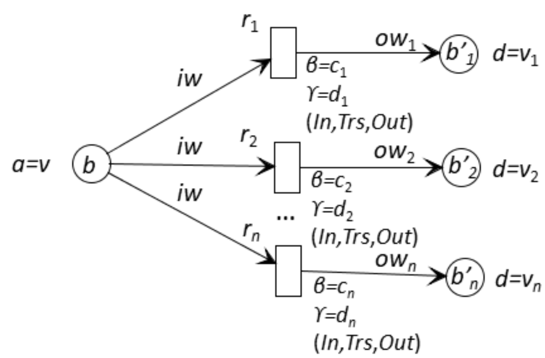

Fig. 4. A WPUP-net model of the type 3 rule after firing r_3.

We assume that for the rules of type 3, In is interpreted as an andlike uninorm.

Remark. In each of the formulas presented above, in the case of nonzero markings of output places, the final value of the token b" should be calculated according to the formula: $b" = Out(b', M(p'))$, where b' is the token value calculated as described above for each of the considered rule types, and $M(p')$ denotes the nonzero marking of the output place p'.

5 An Algorithm

In this section, we introduce an algorithm that describes how a WPUP-net should work. It can be seen that in each computation of the net N_U built on the basis of a given decision table DT, two phases can be distinguished. In the first phase, tokens are set in input places of transitions representing conditional rules, not necessarily in all of them. In its second phase, the algorithm transfers tokens between places on the net as quickly as possible. This phase is implemented by the part of the net that represents all true and acceptable rules in the decision table DT. Net computation ends when the net model has proposed a decision or there are no transitions representing conditional rules ready to fire that have not been used before.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_15

Algorithm 1: WPUP-net model for a given DT.

Input : WPUP-net N_U.
Output: Final decision.
begin
Set the marking of N_U;
A: **if** *decision transitions are ready to fire* **then**
  fire them simultaneously and go to B;
if *condition transitions are ready to fire and they have not been fired yet* **then**
  fire them simultaneously and go to A
else
  Go to C;
B: Read the final decision
C: **end**.

6 An Illustrative Example

In this section, we will illustrate our methodology with the decision table in Example 1 representing a flu diagnosis system.

After transforming the rules with parameters from Example 1 into Petri nets (see Section 4) and combining them in common places, we obtain the resulting WPUP-net shown in Fig. 5. This net contains seven places and ten transitions. Among all places, there are five places (from p_1 to p_5) corresponding to conditional descriptors in the rules (these are starting places; selecting these places starts a diagnosis process), the remaining two refer to decision descriptors (these are treated as decision places whose nonzero tokens indicate the proposed decisions and their calculated degree of credibility). The set of transitions contains five transitions (from t_1 to t_5), which represent conditional rules (from r_1 to r_5), while the rest represent decision rules. Directed arcs connecting places with transitions and vice versa (elements of the sets $P \times T$ and $T \times P$) along with their weights on the arcs are illustrated in Fig. 5. In the initial net marking, p_2 and p_5 contain nonzero tokens, the rest are empty. Place p_2 includes token 3/4 which represents the truth degree of descriptor T = high, and place p_5 contains token 1/2 which represents the truth degree of descriptor H=yes. Statement set S contains all descriptors (conditional and decision) appearing in the set of rules from Example 1. The initial markings of p_2 and p_5 are nonzero, the rest are empty. The marking of p_2 is equal to 3/4 and represents the truth degree of the descriptor T = high, and the marking of p_5 is equal to 1/2 and represents the truth degree of the descriptor H = yes. Statement set S consists of all the descriptors (conditional and decision) that appear in the rule set in Example 1. The elements of S correspond uniquely to their places in the net, as shown in the figure. The role of the degree of truth function β is to assign the value 1 to transitions t_5, t_6, t_7, t_{10}, the value 1/2 to transitions t_1, t_2, t_8, t_9, the value 1/3 to t_3 and finally the value 2/3 to t_4. Threshold function γ assigns 1/10 to each transition. Set of operators O_p contains three operators In, Trs and Out interpreted respectively as uninorms U, U and U' described in Example 2, but

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_15

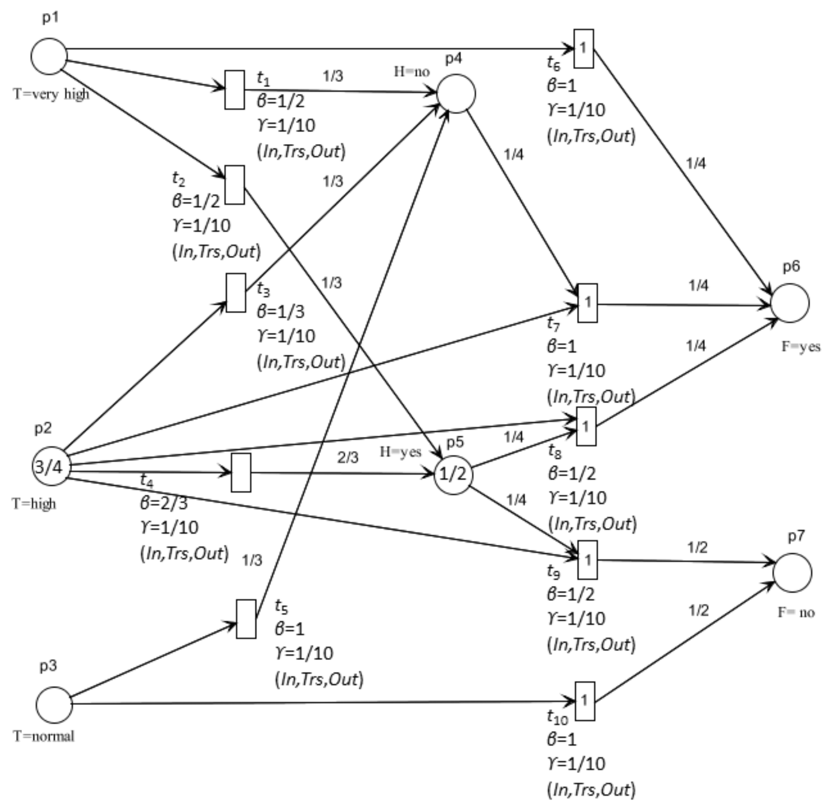

Fig. 5. A WPUP-net model for the rules in Example 1 with the initial marking $M_0 = (0, 3/4, 0, 0, 1/2, 0, 0)$.

now we assume that the neutral element $e = 1/8$. Operator binding function δ assigns the triple form (In, Trs, Out) to each net transition. When evaluating statements attached to p_2 and p_5, we notice that t_3, t_4, t_8 and t_9 are ready to fire in the initial marking. Transitions t_8 and t_9 will run first because their priorities are higher than the priorities of the other two transitions. After firing t_8 and t_9 simultaneously or one at a time in any order, the net operation stops at $1/4$ and $1/2$ corresponding to decisions F = yes, F = no, respectively. Due to the fact that the degree of truth of the statement F = no is greater than the degree of truth of the latter, the net model proposes the statement that there is no flu in the case under consideration.

Now consider the case where only p_2 is marked as before, and the other starting places are empty. It can be checked that in such a situation no transition representing a decision rule is ready to be fired, while t_3 and t_4 are ready. After firing these transitions, we will get a marking at which you can see that t_8 and t_9 are ready to fire. When these transitions are fired simultaneously or one at a time in any order, we get a marking where the net computation ends with a decision proposal indicating no flu and with the same degree of credibility as before. This example shows that the proposed net model can also work in the absence of tokens in some input places of decision transitions in the initial marking, which may result in the inability to make a decision immediately after starting its work. Sometimes missing information is obtained. This is the case when, after starting the model, conditional transitions are ready to fire, which can generate the necessary tokens. This was in our example. Detailed calculations related to the description of net operations have been omitted. They are similar to those described in Example 3 (Section 3).

Remark. All drawings of Petri net models in this paper were made in PNeS [8].

7 Conclusion and Further Work

In the paper, we presented a hybrid methodology that allows you to build concurrent models of decision-making systems based on knowledge extracted from empirical data stored in a given decision table. A new type of Petri net was used to represent the decision-making system for diagnosing flu cases. In this example, the operation of the model was analyzed in terms of the decisions it proposed, with particular emphasis on the situation when the input data of the model did not allow for immediate decision-making due to their incompleteness. The effect of such action is obtained by applying true and acceptable rules in the construction of the model along with the appropriate organization of its work. The expected functioning of the model became possible additionally due to the introduction of differentiated transition priorities and the appropriate interpretation of three transition operators in the uninorm class, responsible for the dynamics of the model's behavior.

Due to the fact that in many real-world situations it is difficult to determine the exact membership value or degree of truth, in further research we intend to focus on interval data rather than exact data [26]. For this purpose, in the WPUP-net model, we intend to replace the classical uninorms with interval uninorms and check experimentally what positive changes both in operation and in the effectiveness and usefulness of the proposed decisions can be obtained. Another problem of interest to us concerns the formulation of requirements under which net models of this type are deterministic (cf. [27]).

Acknowledgement. The authors thank the anonymous reviewers for their helpful comments.

The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Munakata, T.: Fundamentals of the New Artificial Intelligence. Beyond Traditional Paradigms. Springer (1998)

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_15

2. Pawlak, Z.: Rough sets. Int. J. Comput. and Inf. Sci. **11**, 341–356 (1982)
3. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**, 338–353 (1965)
4. Petri, C.A.: Komunikation mit Automaten. Schriften des IIM Nr. **2**. Bonn (1962)
5. Pawlak, Z.: Rough sets - theoretical aspects of reasoning about data. Kluwer (1991)
6. Skowron, A.: Synthesis of adaptive decision systems from experimental data. Artif. Intell. Appl. **28**, 220–238 (1995)
7. Skowron, A., Suraj, Z.: A parallel algorithm for real-time decision making: a rough set approach. J. Intell. Inf. Syst. **7**, 5–28 (1996)
8. Suraj, Z., Grochowalski, P.: PNeS in Modelling, Control and Analysis of Concurrent Systems. Lecture Notes in Artif. Intell. **12872**, pp. 279–309, Springer Nature (2021)
9. Lipp, H.P.: Application of a fuzzy Petri net for controlling complex industrial processes. In: Proc. of IFAC 1984, pp. 471–477 (1984)
10. Cardoso, J., Camargo, H. (eds.): Fuzziness in Petri Nets. Springer (1999)
11. Liu, H.-C., You, J.-X., Li, Z.W., Tian, G.: Fuzzy Petri nets for knowledge representation and reasoning: a literature review. Eng. Appl. Artif. Intell. **60**, 45–56 (2017)
12. Jiang, W., Zhou, K.-Q., Sarkheyli-Hägele, A., Zain, A.M.: Modeling, reasoning, and application of fuzzy Petri net model: a survey. Artif. Intell. Rev. **55**, 6567–6605 (2022)
13. Yager, R.R., Rybalov, A.: Uninorm aggregation operators. Fuzzy Sets Syst. **80**, 111–120 (1996)
14. De Baets, B., Fodor, J.: Van Melle's combining function in MYCIN is a representable uninorms: An alternative proof. Fuzzy Sets Syst. **104**, 133–136 (1999)
15. Yager, R.: Uninorms in fuzzy systems modeling. Fuzzy Sets Syst. **122**, 167–175 (2001)
16. Mesiarová, A.: Multi-polar t-conorms and uninorms. Inform. Sci. **301**, 227–240 (2015)
17. Yager, R. R., Rybalov, A.: Bipolar aggregation using the uninorms. Fuzzy Optim. Decis. Making **10**, 59–70 (2011)
18. Suraj, Z.: A Hybrid Approach to Approximate Real-time Decision Making. In: Proc. of FUZZ-IEEE 2021, Luxembourg, pp. 71–78, IEEE (2021)
19. Dubois, D., Prade, H.: A review of fuzzy sets aggregation connectives. Inform. Sci. **36**, 85–121 (1985)
20. Li, Y., Shi, Z.: Remarks on uninorms aggregation operators. Fuzzy Sets Syst. **114**, 377–380 (2000)
21. Klement, E.P., Mesiar, R., Pap, E.: Triangular norms. Kluwer (2000)
22. Pawlak Z., Skowron, A.: Rudiments of rough sets. Inf. Sci. **177**, 3–27 (2007)
23. Fodor, J.C., Yager, R.R., Rybalov, A.: Structure of uninorms. Int. J. Uncertain Fuzziness Knowl.-Based Syst. **5**, 411–427 (1997)
24. Hack, M.: Decidability Questions for Petri Nets, Ph.D. dissertation. MIT, Cambridge, Massachusetts (1975)
25. Ha, M.H., Li, Y., Wang, X.F.: Fuzzy knowledge representation and reasoning using a generalized fuzzy Petri net and a similarity measure. Soft Comput. **11**(4), 323–327 (2007)
26. Moore, R.E., Kearfott, R.B., Cloud, M.J.: Introduction to Interval Analysis. SIAM: Philadelphia, PA, USA (2009)
27. Suraj, Z.: On Selected Properties of Uninorm Petri Nets and Their Application in Modeling Knowledge-Based Systems. Procedia Comput. Sci. **225**, 155–164 (2023)

DOI: 10.1007/978-3-031-63751-3_15